



برنامه نویسی به زبان C++



عطیه پوردرخشان • لادن جاماسبی



مجموعه کتاب‌های علامه حلی

برنامه نویسی سی پلاس پلاس ۲

- لادن جاماسبی
- عطیه پوردرخشان





سرشناسه : جاماسبی، لادن، ۱۳۷۸
عنوان و نام پدیدآور : برنامه نویسی، سی پلاس پلاس (۱) / لادن جاماسبی، عطیه پوردرخشان.
مشخصات نشر : تهران: انتشارات حلی: دانش پژوهان جوان، ۱۴۰۰.
مشخصات ظاهری : ۹۲ ص.: مصور (بخشی رنگی)
فروست : مجموعه کتاب‌های علامه حلی
شابک : ۹۷۸-۶۰۰-۴۹۶-۲۲۵-۴
وضعیت فهرست نویسی : فیپا
موضوع : سی ++ (زبان برنامه نویسی کامپیوتر)
موضوع : ++ (Computer program language) C++
زبان‌های برنامه نویسی کامپیوتر
Programming language (Electronic computers)
شناسه افزوده : پوردرخشان، عطیه، ۱۳۷۰-
رده‌بندی کنگره : QA ۷۶/۷۳
رده‌بندی دیویی : ۰۵/۱۳۳
شماره کتابشناسی ملی : ۸۶۸۳۰۴۱

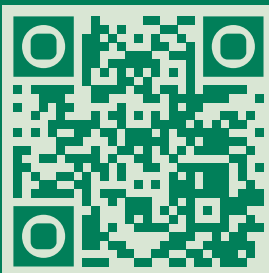


برنامه نویسی سی پلاس پلاس ۲
انتشارات حلی
انتشارات دانش پژوهان جوان
لادن جاماسبی، عطیه پوردرخشان
محمد رضا معتبر
سمیه سادات فاطمی
راضیه فرهانیان
زهرة شیروانی هرندي
محمد حسین صفدریان
۱۴۰۱
اول
واژه پرداز اندیشه
۲۰۰۰ جلد
۶۳۰۰۰ تومان
۹۷۸-۶۰۰-۴۹۶-۲۲۵-۴

عنوان کتاب
ناشر
ناشر همکار
مؤلفان
ویراستار علمی
مسئول هماهنگی
صفحه آرا
طراح جلد
تصویر ساز
سال چاپ
نوبت چاپ
چاپ و صحافی
شمارگان
قیمت
شماره شابک



تهران، خیابان انقلاب، میدان فردوسی، ابتدای کوچه براتی، پلاک ۱۶ و ۱۴
تلفن دفتر مرکزی: ۶۶۷۴۴۳۸۴-۴









بهترین روش برای یادگیری عمیق و کاربردی برنامه نویسی، دست به کد شدن و حل تمرین‌های متنوع است. مخصوصاً اگر تمرین‌ها در کمترین لحظه ممکن صحیح شوند و بتوانیم درستی و نادرستی آن را در کوتاه‌ترین زمان متوجه شویم و سعی کنیم اشکالات خود را برطرف کنیم. سامانه آموزش برنامه نویسی کوئرا، مجهز به داوری خودکار سوالات و تمرین‌هاست و به کمک آن می‌توانید از صحت سوالات حل شده خود مطلع شوید.
برای شروع به سایت Quera.org بروید، یک حساب کاربری برای خود باز کنید و با اسکن این بارکد، به صفحه اصلی کتاب وارد شوید و با انتخاب فصل مورد نظر، تمرین خود را انتخاب و پاسخ آن را ارسال کنید.

کلیه حقوق این اثر برای ناشر محفوظ است.

هیچ شخص حقیقی یا حقوقی حق برداشت تمام یا قسمتی از اثر را به صورت چاپ، فتوکپی، جزوه و مجازی ندارد.

متخلفان به موجب بند ۵ از ماده ۲ قانون حمایت از ناشران تحت پیگرد قانونی قرار می‌گیرند.



فصل ۰ مرور مباحث گذشته	درسنامه ۵	تمرین ۱۴	
فصل ۱ حلقه شمارشی	درسنامه ۱۵	تمرین ۲۸	
فصل ۲ حلقه‌های تودرتو	درسنامه ۲۹	تمرین ۳۸	
فصل ۳ آرایه‌ها	درسنامه ۳۹	تمرین ۵۲	
فصل ۴ رشته‌ها	درسنامه ۵۳	تمرین ۶۲	
فصل ۵ تابع	درسنامه ۶۳	تمرین ۷۶	

قبل از شروع به مطالعه کتاب، این قسمت را بخوانید:

وقتی شروع به خواندن این کتاب کنید با بخش‌های مختلفی مواجه می‌شوید که غالباً یک لاک‌پشت متفاوت در اول هرکدام وجود دارد. برای هرکدام از این بخش‌ها از شما انتظار داریم کار متفاوتی انجام دهید. این قسمت‌ها بر اساس تئوری‌های نوین آموزش و تجارب موفق تدریس برای آموزش دانش‌آموزان مستعد طراحی شده است. این بخش‌ها شامل:

درخت دانش: در صفحه دوم هر فصل، نمودار دایره‌ای شکلی کشیده شده که به ما کمک می‌کند بفهمیم در آن فصل مطالب علمی چطور تقسیم‌بندی شده و ارتباط آن‌ها با هم چیست. درواقع این بخش نقشه‌ای است برای گم نشدن در موضوعات علمی.

اهداف رفتاری: زیر هر درخت دانش، چند جمله نوشته شده که از اول کار معلوم کند که این فصل را می‌خوانیم چه بشود. خوب است در آخر فصل هم برگردیم و ببینیم که می‌توانیم کارهایی را که در این بخش گفته انجام دهیم یا نه.

پاسخگو باش: در این قسمت باید پاسخگو باشیم. پاسخگوی سؤالی که پرسیده شده و انتظار می‌رود بعد از خواندن درس تا آن قسمت، بتوانیم باکمی فکر کردن به آن جواب دهیم.

فسفر بسوزان: شاید لازم باشد مقدار بیشتری از مغز خودمان استفاده کنیم و قدری از فسفرهای ذخیره‌شده را بسوزانیم! سؤالاتی که در بخش فسفر بسوزان مطرح می‌شود فقط با خواندن مطالب درسی قابل پاسخگویی نیست و باید کمی بیش از معمول درباره آن‌ها فکر کنیم.

جالب است بدانی: برای افرادی که دوست دارند بیشتر از سطح استاندارد با موضوعات آشنا شوند این قسمت توصیه می‌شود. در این قسمت مطالبی آورده شده که خواندن و یادگرفتن آن الزامی نیست ولی آن‌قدر جذاب است که نشود به راحتی بی‌خیال خواندن آن شد.

لغت‌نامه: ما دانش‌آموزان مستعد و متفاوت (!) دوست داریم بتوانیم علاوه بر مطالب درسی، جست‌وجویی هم بکنیم و ببینیم در دنیا درباره موضوع درسی ما چه چیزی وجود دارد. برای همین در پایان هر فصل لغات مهم فصل با معادل انگلیسی آن آورده شده است.

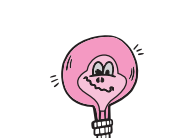
تمرین‌ها: در آخر هر فصل تمرین‌های مرتبط با آن آورده شده است. از آنجایی که مؤلفان کتاب از دبیران باسابقه هستند پس تعداد تمرین‌ها، وقت لازم برای انجام آن‌ها، تعداد سؤالات سخت و آسان و نوع سؤالات با برنامه و محاسبه تعیین شده است. پس خیالتان راحت باشد که همه تمرین‌ها را در طول سال می‌شود انجام داد. تمرین‌ها بر اساس موضوعات هر فصل بخش‌بندی شده، بنابراین لازم نیست برای تمرین منتظر پایان فصل باشید؛ در پایان هر مبحث می‌توانید به بخش تمرین‌ها مراجعه کنید و تمرین‌های همان مبحث را حل کنید.

دست‌به‌کد شو: برنامه‌نویسی درسی کاربردی است که در حین آموزش آن لازم است شما هم دست به کد بشوید. در بخش دست به کد شو از شما خواسته شده تا سعی کنید خودتان برنامه را بنویسید. حواستان باشد این بخش، قسمت مهمی از روند درسی است و نمی‌شود بدون دست‌به‌کد شدن برنامه نویسی یاد گرفت.

اشتباه رایج: همان‌طور که از اسمش مشخص است، در این قسمت اشتباهاتی که ممکن است برای هرکسی پیش آید را برای شما توضیح داده‌ایم تا شما دیگر آن‌ها را تکرار نکنید! می‌توان گفت، اگر قرار باشد در بخش‌های دیگر راه برنامه‌نویسی را یاد بگیرید، در این قسمت با چاه‌های آن آشنا می‌شوید.

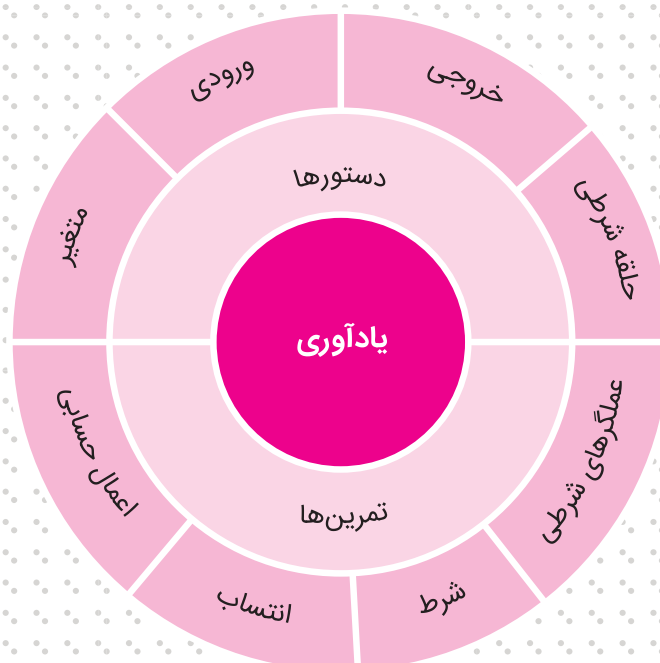
چه می‌کنه: در این قسمت، یک برنامه کامل برای شما نوشته‌ایم و از شما انتظار داریم بگویید این برنامه برای چه هدفی نوشته شده، چه کاری انجام می‌دهد و برای ورودی‌های مختلف، چه خروجی تولید می‌کند.

در ضمن شما هم می‌توانید برای ما مطالب و مسئله ارسال کنید! مطالب و مسئله‌هایی که خودتان از آن‌ها لذت برده‌اید به آدرس: ketab.helli@gmail.com





فصل صفر مرور مباحث گذشته



اگر این فصل را به خوبی مطالعه کنی و کارهای خواسته شده را به دقت انجام دهی:

- آغاز برنامه نویسی به زبان C++ را به همراه مباحث ورودی و خروجی، متغیر، اعمال حسابی، شرط و حلقه های شرطی را مرور می کنی.



محیط برنامه نویسی

برای نوشتن یک برنامه به زبان C++ به یک محیط توسعه یکپارچه یا IDE نیاز داریم که برای این زبان قابل استفاده باشد. این محیط شامل ویرایشگر کد برای نوشتن دستورات، کامپایلر یا مفسر برای ترجمه دستورات نوشته شده به زبان کامپیوتر (زبان صفر و یک) و اصلاح کننده کد که در رفع اشکالات برنامه به ما کمک می کند و خطاهای موجود در برنامه را برایمان مشخص خواهد کرد.

چند نمونه IDE مناسب برای زبان C++ نام ببرید و تحقیق کنید کدام یک از آن ها می تواند برای شما مناسب تر باشد.



کنکاش کن

آغاز برنامه نویسی به زبان C++

با برنامه زیر آغاز می کنیم:

```
#include<iostream>
using namespace std;

int main()
{
    int n;
    cout << "salam! lotfan adad vared konid ..." << endl;
    cin >> n;
    cout << "eyval :)" << endl;
    cout << "adad voroodi shoma : " << n << endl;
}
```

خطوط ابتدایی: کتابخانه ها

هر برنامه به زبان C++ در خطوط ابتدایی خود یک یا چند include را شامل شده است. این خطوط نشان دهنده نام کتابخانه هایی هستند که ما می خواهیم از دستورات آن ها استفاده کنیم و در حقیقت مجوز استفاده از دستورات هر کتابخانه را با نوشتن نام آن در ابتدای برنامه از کامپایلر دریافت می کنیم. در این برنامه از کتابخانه `iostream` نام برده شده که مجوز استفاده از دستورات ورودی و خروجی را برایمان صادر می کند. این را هم بدانیم که پس از نوشتن تمام کتابخانه های مورد نظر برای برنامه می بایست عبارت `using namespace std` را در تمام برنامه هایمان داشته باشیم.

اصل مطلب : `int main`

مهم ترین بخش برنامه، در این قسمت؛ یعنی در `int main` اتفاق می افتد. وقتی برنامه به این قسمت می رسد، می داند که باید منتظر دستورات ما باشد.

دستورات

در خط ششم برنامه یک متغیر از نوع عدد صحیح به نام `n` تعریف کرده ایم. در خط هفتم با استفاده از دستور `cout` عبارت مورد نظرمان را چاپ کرده ایم و به کمک دستور `endl` اشاره کرده ایم چاپ خروجی به خط بعدی

می‌رود. بعد از آن دستور cin نوشته شده که به وسیله آن مقدار متغیر n را از کاربر دریافت می‌کنیم و در نهایت در دو خط انتهایی مقدار دریافت شده همراه یک عبارت مناسب چاپ می‌شود.

متغیر

متغیرها ظرف‌هایی از حافظه هستند که اطلاعاتی مانند اعداد، حروف و ... را در خود ذخیره می‌کنند. این ظروف، ظرفیت محدود و مشخصی دارند و به کمک نامی که برای آن‌ها انتخاب می‌کنیم، می‌توانیم از آن‌ها استفاده کنیم.

با سه نوع مهم از متغیرها در سی‌پلاس‌پلاس آشنا شدیم :

اعداد صحیح (int)

اعداد اعشاری (double)

کاراکتر (char)



تعریف متغیر به نام a از جنس عدد صحیح و مقداردهی به آن :

```
int a = 5;
```

با این دستور یک بخش از حافظه که می‌تواند حاوی یک عدد صحیح باشد به متغیر a اختصاص داده می‌شود و مقدار ۵ در آن قرار می‌گیرد.



تعریف متغیر به نام num از جنس عدد اعشاری :

```
double num;
```

با این دستور یک بخش از حافظه که می‌تواند شامل یک عدد اعشاری باشد به متغیر num اختصاص پیدا می‌کند.



تعریف متغیر به نام harf از جنس حرف و علامت:

```
char harf;
```

با این دستور یک بخش از حافظه که می‌تواند حاوی یک حرف یا یک علامت باشد به متغیر harf اختصاص داده می‌شود.

اعمال حسابی

به‌طور کلی به عملیات اصلی ریاضی؛ یعنی جمع، تفریق، تقسیم و ضرب، چهار عمل اصلی می‌گوییم. جالب است بدانید که این اعمال، در برنامه‌نویسی هم وجود دارند و از آن‌ها بسیار زیاد استفاده می‌کنیم.

عملیات	نحوه نوشتن با متغیرهای a و b
جمع	a+b
تفریق	a-b
ضرب	a*b
تقسیم	a/b

مثال:

```
#include <iostream>
using namespace std;

int main()
{
    int u = 7, x = 3;
    cout << u - x;
}
```

پس از اجرای این برنامه در صفحه نمایش عدد ۴ نمایش داده می‌شود.

برنامه‌ای بنویسید که دو عدد اعشاری از ورودی دریافت کرده و در خروجی به همراه پیغام‌های مناسب به ترتیب، مجموع، ضرب و میانگین آن‌ها را چاپ نماید.



دست به کار
شو

پیدا کردن باقی‌مانده تقسیم، به کمک یک عملگر خاص انجام می‌شود. برای به دست آوردن باقی‌مانده تقسیم دو عدد صحیح، کافی است از نماد % استفاده کنیم.

عملیات باقی‌مانده	برای دو متغیر صحیح a و b
%	a/b

برنامه‌ای بنویسید که در ورودی یک عدد طبیعی دریافت کرده و یکان آن عدد را چاپ کند.

برنامه‌ای بنویسید که در ورودی یک عدد طبیعی دریافت کرده و دهگان آن عدد را چاپ کند.



دست به کار
شو

در برنامه‌نویسی برای هریک از حروف، یک عدد منحصر به فرد داریم که اگر آن عدد را به صفر و یک تبدیل کنیم کامپیوتر متوجه یک کاراکتر خواهد شد. این اعداد و حروف متناظرشان، در دنیای برنامه‌نویسی در جدولی به نام جدول کد اسکی نوشته شده‌اند.

سوالاتی که وجود دارد این است که چطور کد اسکی یک کاراکتر را تشخیص دهیم. در پاسخ خوب است بدانیم کد اسکی یک حرف، از نوع عدد صحیح است پس برای پیدا کردن کد اسکی یک حرف، می‌توانیم از روش زیر استفاده کنیم:

```
#include <iostream>
using namespace std;

int main()
{
    char a = 'p';
    int x;
    x = a;
    cout << "a : " << x << endl;
}
```

شرط

```
if (شرط ۱)
{
    دستور ۱;
    دستور ۲;
    دستور ۳;
    ...
}
else if (شرط ۲)
{
    دستور ۱;
    دستور ۲;
    دستور ۳;
    ...
}
else
{
    دستور ۱;
    دستور ۲;
    دستور ۳;
    ...
}
}a
```



اگر شرط‌های بیشتری داشتیم، می‌توانیم برای هر if به تعداد مورد نیاز else if داشته باشیم.

برای نوشتن شرط‌ها به آشنایی با عملگرهای مقایسه‌ای نیاز داریم:

در ریاضی	در C++
>	>
<	<
=	==
≠	!=
≤	<=
≥	>=



پایه می گذاریم

```
#include <iostream>
using namespace std;

int main()
{
    char ch;
    cin >> ch;
    if (ch == 's')
    {
        cout << "Salam!" << endl;
    }
    else if (ch == 'h')
    {
        cout << "Hello!" << endl;
    }
    else if (ch == 'w')
    {
        cout << "Welcome!" << endl;
    }
    else
    {
        cout << "What do you mean?" << endl;
    }
}
```

برنامه‌ای بنویسید که مطابق جدول زیر باتوجه به دمای آب، حالت آن را گزارش دهد. ورودی شامل یک عدد نشان‌دهنده دمای آب و خروجی شامل یک عبارت نشان‌دهنده حالت آب خواهد بود.



دست به کار
شو

دمای آب	حالت آب
کمتر مساوی صفر	Ice (یخ)
بین صفر و صد	Water (آب)
بیشتر از ۱۰۰	Steam (بخار)

نوشتن شرط‌های تودرتو کار پرخطری است! رعایت نکردن نکات جزئی می‌تواند منجر به خطاهای منطقی در برنامه شود. یکی از گزینه‌هایی که در برخی موارد می‌توانیم جایگزین شرط‌های تودرتو کنیم، استفاده از عملگرهای منطقی است. به کمک عملگرهای منطقی، می‌توانیم به راحتی شرط‌ها را با هم ترکیب کنیم. فقط باید به یک نکته توجه داشته باشیم، اینکه چه زمانی شرط درست و چه زمانی نادرست است. سه عملگر منطقی (و)، (یا) و (نقیض یا نه) را معرفی می‌کنیم:

و:

فرض کنید P و Q دو شرط باشند، مقدار $P \&\& Q$ در صورتی درست است که هر دو شرط درست باشند؛ یعنی هم P و هم Q درست باشد. وگرنه درست نیست.

پله می‌گذرد؟



```
#include <iostream>
using namespace std;

int main()
{
    int a, b;
    cin >> a >> b;
    if ((a > 0) && (b > 0))
    {
        cout << a * b << endl;
    }
}
```

یا:

فرض کنید P و Q دو شرط باشند، مقدار $P || Q$ در صورتی درست است که یکی از دو شرط درست باشند؛ یعنی یا P یا Q یا هر دو درست باشند وگرنه درست نیست.

پله می‌گذرد؟



```
#include <iostream>
using namespace std;

int main()
{
    if (1 == 1 || 1 > 2)
    {
        cout << "man chap misham?" << endl;
    }
}
```

نه:

این عملگر منطقی، مثل دیوی در کلاه قرمزی کار می‌کند؛ یعنی اگر پشت هر شرطی بیاید، آن را برعکس می‌کند، اگر درست باشد آن را نادرست، اگر نادرست باشد درستش می‌کند.

```
#include<iostream>
using namespace std;

int main()
{
    int a, b;
    cin >> a >> b;
    if(!(a + b >= 10))
        cout << "a + b kamtar az 10 ast";
    else
        cout << "a + b kamtar az 10 nist";
}
```



پایه می‌کند



توجه کنید که زمانی که if، elseif یا else تنها شامل یک خط دستور باشند می‌توان از نوشتن { } صرف‌نظر کرد و در این صورت خط دستور مورد نظر مربوط به if، else یا elseif بالای خود در نظر گرفته می‌شود.



حلقه شرطی

حلقه while، دستوری است که می‌توان به کمک آن، دستورات را تا زمانی که شرطی برقرار باشد اجرا کرد. برای اینکه بتوانیم از این دستور استفاده کنیم، کافی است مانند نمونه، آن را در برنامه خود به کار ببریم تا بتوانیم دستوراتمان را با شرط دلخواه، تکرار کنیم.

```
While (شرط)
{
    دستور ۱;
    دستور ۲;
    دستور ۳;
    ...
}
```

برنامه زیر اعداد ۱ تا ۱۰۰ را در خطوط متفاوت خروجی چاپ می‌کند:


```
#include<iostream>
using namespace std;

int main()
{
    int t = 1;
    while(t <= 100)
    {
        cout << t << endl;
        t = t + 1;
    }
}
```



برنامه‌ای بنویسید که یک عدد صحیح بزرگ‌تر از یک دریافت کرده و در خروجی مجموع مقسوم‌علیه‌های آن را چاپ کند.

اگر در یک حلقه تکرار به دستور **break** برسیم کار حلقه همان‌جا تمام شده و برنامه از حلقه خارج می‌شود.



```
#include<iostream>
using namespace std;

int main()
{
    int n = 1;
    while(1 == 1)
    {
        cout << n << endl;
        if(n == 10)
            break;
        n++;
    }
}
```

اگر در یک حلقه تکرار به دستور **continue** برسیم ادامه دستورات پس از آنکه در حلقه موجود هستند اجرا نشده و اجرای حلقه به مرحله یا دفعه بعدی خود می‌رود و شرط حلقه چک می‌شود.



```
#include<iostream>
using namespace std;

int main()
{
    int n, t = 0, x = 0;
    cin >> n;
    while(x < n)
    {
        x++;
        if(n % x != 0)
            continue;
        t++;
    }
    cout << t;
}
```

تمرین‌ها



۱. برنامه‌ای بنویسید که دو عدد از کاربر گرفته و در متغیرهایی به نام a و b ذخیره کند سپس محتوای این دو متغیر را با هم جابه‌جا کند!
۲. برنامه‌ای بنویسید که n عدد از ورودی دریافت کرده و بزرگ‌ترین آن‌ها را چاپ کند.
۳. یک برنامه بنویسید که عدد طبیعی n را از کاربر بگیرد و بگوید که این عدد بر مجموع ارقامش بخش‌پذیر هست یا خیر.
۴. برنامه‌ای بنویسید که اندازه اضلاع سه ضلع یک مثلث را بگیرد و بگوید که مثلث متساوی‌الساقین است یا خیر.
۵. برنامه‌ای بنویسید که یک کاراکتر از ورودی بگیرد، اگر این کاراکتر از حروف انگلیسی نبود در خروجی علامت تعجب چاپ کند، اگر این کاراکتر از حروف بزرگ انگلیسی بود، حرف کوچک آن را چاپ کند و اگر حرف کوچک انگلیسی بود در خروجی حرف بزرگ آن را چاپ نماید! (راهنمایی: از کد اسکی استفاده نمایید).

آشنایی با سوابق آموزش تیم تالیف

عطیه پوردرخشان:

- مدرس برنامه نویسی دبیرستان انرژی اتمی
- مسئول و طراح دوره آموزشی کدآپ



لادن جاماسبی:

- مدرس برنامه نویسی دبیرستان فرزندگان یک تهران
- عضو گروه آموزشی کدآپ



۰۲۱-۶۶ ۷۳ ۷۳ ۳۸
www.mhelli.ir
@mhelli7-8-9

۰۲۱-۶۶ ۴۹۶۳۶۳۳
www.dpjpub.com
@dpjpub

